Agent Based Software Engineering

Activity

Name : Muhammad Shamoil
Registration ID : 232025
Section : BSSE - V – A
Submitted To : Sir Anwaar

1. Upgraded Automatic Door

```python
def upgraded_automatic_door(motion_detected, person_standing_near):

    if motion_detected:
        print("Door Opens")
    elif not motion_detected and person_standing_near:
        print("Door Remains Open")
    else:
        print("Door Closes")

# ---------------------------------------------- TEST CASES -------------
print("## 1. Upgraded Automatic Door ##")
upgraded_automatic_door(True, True)    # Motion detected
upgraded_automatic_door(False, True)   # No motion, but person is near
upgraded_automatic_door(False, False)  # No motion, no one near
```

```
## 1. Upgraded Automatic Door ##
Door Opens
Door Remains Open
Door Closes
```

2. Enhanced Traffic Light

```python
def enhanced_traffic_light(car_detected, pedestrian_waiting):

    if car_detected and pedestrian_waiting:
        print("Light is RED (Longer for pedestrians)")
    elif car_detected:
        print("Light is RED (Normal duration)")
    else:
        print("Light is GREEN")

# ---------------------------------------------- TEST CASES -------------

print("## 2. Enhanced Traffic Light ##")
enhanced_traffic_light(True, True)    # Car and pedestrian detected
enhanced_traffic_light(True, False)   # Only car detected
enhanced_traffic_light(False, False)  # No car or pedestrian
```

```
## 2. Enhanced Traffic Light ##
Light is RED (Longer for pedestrians)
Light is RED (Normal duration)
Light is GREEN
```

3. Smart Street Light

```python
def smart_street_light(time, motion, weather):

    if weather == "rainy":
        print("Weather is rainy. Street Light ON")
    elif time == "night":
        if motion:
            print("Night time with motion. Street Light is BRIGHT")
        else:
            print("Night time with no motion. Street Light is DIM")
    else:
        print("Day time, clear weather. Street Light OFF")

# ---------------------------------------------- TEST CASES -------------
print("## 3. Smart Street Light ##")
smart_street_light("day", False, "rainy")  # Rainy day
smart_street_light("night", True, "clear")   # Motion at night
smart_street_light("night", False, "clear")  # No motion at night
smart_street_light("day", False, "clear")    # Clear day
```

```
## 3. Smart Street Light ##
Weather is rainy. Street Light ON
Night time with motion. Street Light is BRIGHT
Night time with no motion. Street Light is DIM
Day time, clear weather. Street Light OFF
```

4. Extended Temperature Control

```python
def extended_temp_control(temp, humidity):

    # Temperature control
    if temp < 18:
        print("Heater ON")
    elif temp > 26:
        print("AC ON")
    else:
        print("Temperature is OK")

    # Humidity control
    if humidity > 60:
        print("High humidity detected. Dehumidifier ON")
    else:
        print("Humidity is OK")

# ---------------------------------------------- TEST CASES -------------
print("4. Extended Temperature Controller ")
extended_temp_control(15, 70) # Too cold and humid
extended_temp_control(30, 50) # Too hot, humidity ok
extended_temp_control(21, 65) # Temp ok, too humid
```

```
4. Extended Temperature Controller
Heater ON
High humidity detected. Dehumidifier ON
AC ON
Humidity is OK
Temperature is OK
High humidity detected. Dehumidifier ON
```

5. Extended Sanitizer Dispenser

```python
def extended_sanitizer_dispenser(hand_detected, level):

    if level <= 10:
        print("ALERT: Sanitizer level is low! Please refill.")
    elif hand_detected:
        print("Dispensing Sanitizer")
    else:
        print("Waiting for Hand...")

# ---------------------------------------------- TEST CASES -------------
print("5. Extended Sanitizer Dispenser")
extended_sanitizer_dispenser(True, 5)    # Hand detected, but level is low
extended_sanitizer_dispenser(True, 50)   # Hand detected, level is OK
extended_sanitizer_dispenser(False, 50)  # No hand detected
```

```
5. Extended Sanitizer Dispenser
ALERT: Sanitizer level is low! Please refill.
Dispensing Sanitizer
Waiting for Hand...
```

6.Extended Seat belt Alarm

```python
def extended_seatbelt_alarm(seatbelt_fastened, car_moving):
    """
    Sounds an alarm only if the seatbelt is unfastened AND the car is moving.
    """
    if not seatbelt_fastened and car_moving:
        print("ALARM: Please Fasten Your Seatbelt!")
    elif not seatbelt_fastened and not car_moving:
        print("Reminder: Seatbelt is not fastened.")
    else:
        print("Seatbelt Fastened. Safe to drive.")

# ---------------------------------------------- TEST CASES -------------
print("6. Extended Seatbelt Alarm")
extended_seatbelt_alarm(False, True)   # Not fastened and moving
extended_seatbelt_alarm(False, False)  # Not fastened and stationary
extended_seatbelt_alarm(True, True)    # Fastened and moving
```

```
6. Extended Seatbelt Alarm
ALARM: Please Fasten Your Seatbelt!
Reminder: Seatbelt is not fastened.
Seatbelt Fastened. Safe to drive.
```

7. Extended Smart Thermostat

```python
class ExtendedSmartThermostat:
    def __init__(self):
        self.previous_temp = None
        self.ideal_range = (19, 25)

    def update_state(self, current_temp):
        if self.previous_temp is None:
            print(f"Setting initial temperature to {current_temp}°C...")
        elif self.ideal_range[0] <= current_temp <= self.ideal_range[1]:
            print(f"Temperature {current_temp}°C is ideal. Turning systems OFF.")
        elif current_temp > self.previous_temp and current_temp > self.ideal_range[1]:
            print(f"It's getting hotter ({current_temp}°C)! Turning on the AC.")
        elif current_temp < self.previous_temp and current_temp < self.ideal_range[0]:
            print(f"It's cooling down ({current_temp}°C)! Turning on the heater.")
        else:
            print(f"Temperature is steady at {current_temp}°C.")
        self.previous_temp = current_temp

# ---------------------------------------------- TEST CASES -------------
print("7. Extended Smart Thermostat")
thermostat = ExtendedSmartThermostat()
thermostat.update_state(28) # Initial setup -> Hot
thermostat.update_state(22) # Cooling down into ideal range
thermostat.update_state(15) # Cooling down -> Heater ON
```

```
7. Extended Smart Thermostat
Setting initial temperature to 28°C...
Temperature 22°C is ideal. Turning systems OFF.
It's cooling down (15°C)! Turning on the heater.
```

8.Extened Water Tank

```python
class ExtendedWaterTank:
    def __init__(self, level=50):
        self.level = level

    def fill(self, amount=15):
        if self.level >= 95:
            print(f"Water level is {self.level}%. Draining automatically to prevent overflow.")
            self.drain(20)
        else:
            self.level += amount
            print(f"Filling... Current level: {self.level}%")

    def drain(self, amount=15):
        if self.level < 20:
            print(f"WARNING: Low water at {self.level}%. Cannot drain further.")
        else:
            self.level -= amount
            print(f"Draining... Current level: {self.level}%")

# ---------------------------------------------- TEST CASES -------------
print("8. Extended Water Tank System")
tank = ExtendedWaterTank(85)
tank.fill() # Fills to 100
tank.fill() # Tries to fill more, but auto-drains first
tank.drain(70) # Drains to 10
tank.drain() # Tries to drain, but level is too low
```

```
8. Extended Water Tank System
Filling... Current level: 100%
Water level is 100%. Draining automatically to prevent overflow.
Draining... Current level: 80%
Draining... Current level: 10%
WARNING: Low water at 10%. Cannot drain further.
```

9.Extened Battery System

```python
class ExtendedBattery:
    def __init__(self, charge=80):
        self.charge = charge

    def use(self, amount=25):
        if self.charge <= 10:
            print(f"CRITICAL BATTERY ({self.charge}%). Auto-powering off.")
        else:
            self.charge -= amount
            print(f"Using battery... Current charge: {self.charge}%")

    def recharge(self, amount=25):
        if self.charge >= 100:
            print(f"WARNING: Battery is full ({self.charge}%). Please unplug.")
        else:
            self.charge += amount
            if self.charge > 100: # Cap charge at 100
                self.charge = 100
            print(f"Recharging... Current charge: {self.charge}%")


# ---------------------------------------------- TEST CASES -------------
print("## 9. Extended Battery System ##")
battery = ExtendedBattery(90)
battery.recharge() # Recharges battery back to 100
battery.recharge() # Get Warning that battery is full
battery.use(95)    # Drains to 5
battery.use()      # Critical battery, powers off
```

```
## 9. Extended Battery System ##
Recharging... Current charge: 100%
WARNING: Battery is full (100%). Please unplug.
Using battery... Current charge: 5%
CRITICAL BATTERY (5%). Auto-powering off.
```